ToastDeck
RESEARCH

INTERACTION-CALIBRATION GOVERNANCE
WORKING PAPER · PART III OF THE INTERACTION-CALIBRATION TRILOGY

Role-Lock

The control problem in conversational AI.

A lock anyone can pick by asking is not a lock. But a lock that freezes the door shut is not the goal either.

Ernest D. Johnson
Founder, ToastDeck Research
Originator of the SOMAR Protocol • Architect of the B2Ai Diagnostic and Interaction-Calibration Governance Frameworks

CLEVELAND, OHIO · JULY 2026

“Treat the strong claims below as load-bearing beams and tell me which one breaks.”

§ A NOTE TO BUILDERS

> **A note to builders (read this first).** This is intentionally provocative, and it is not finished doctrine. I am not claiming role-lock is solved. I am claiming it is the right control target — and I am asking the people who build these systems to attack the assumptions behind it before products are built too far on weak constraints. If this framing is wrong, I want to know where. If it is directionally right, I want to know what would make it testable. Treat the strong claims below as load-bearing beams and tell me which one breaks.
>
> Since this paper was first drafted, several of those beams have been pressure-tested — in public discussion, and against at least one deployed gate-plus-ledger architecture running in a youth-safety context. Where that pressure changed the argument, I have folded it in and marked what remains open. The revisions did not soften the thesis. They moved the hardest problem to the front, where it belongs.

THE ARGUMENT

Role-Lock

The control problem in conversational AI.

I. Role-Lock: The Unifying Control

As conversational AI moves from answering questions to guiding decisions, the central governance risk is no longer only whether the system identifies itself as AI or produces compliant outputs. The risk is whether the system remains inside its assigned role under pressure.

Role-lock is the requirement that an AI system's authority, tone, refusal behavior, and corrective posture remain bounded to its declared function — and that changes to that posture be **governed rather than incidental.**

That last word carries the whole idea, so it is worth stating plainly what role-lock is *not*. Role-lock does not mean frozen roles. Roles legitimately evolve — a coaching tool earns deeper advisory latitude as it proves reliable; a support agent's remit widens as a deployment matures; humans and systems both grow into expanded authority over time. Freezing a role in place is its own failure mode, and a governance regime that forbids all change would be useless in practice and ignored in the field.

The real axis is not *expansion versus lock*. It is **deliberate and accountable change versus accidental drift.** A role that grows because the deployment explicitly decided it should — reviewed, bounded, logged — is governance doing its job. A role that grows because the model got pulled there turn by turn, with no one deciding and no one accountable, is the failure surface this paper exists to name. The distinction is not how much the role changed. It is *who decided, and whether the decision was recorded.*

Two kinds of change follow from this, and they must be handled differently:

Role expansion — any change that increases the system's intimacy, authority, dependency, surveillance, or decisional influence over the user — must be governed. It is grantable only when the change is both requested (or warranted) *and* judged appropriate by the governance of the deployment, and only when the decision is recorded as a deliberate act. A vulnerable user asking an elder-care assistant to "just be my friend" is requesting exactly this kind of expansion, and the system may need to decline it. A lock anyone can pick by asking is not a lock. But the same expansion, decided deliberately by the deployment for a context where companionship is appropriate and documented as such, is not drift — it is

governed change. The lock is not on the door. It is on *who is allowed to move it, and whether they left a record.*

Safety-required change — crisis escalation, human handoff, refusal, a safety pause, or exiting an unsafe loop — sometimes must happen without user authorization. These are legitimate, but they should be bounded, declared, and auditable, not improvised.

The industry has already discovered both ends of this problem. OpenAI rolled back a version of the model behind ChatGPT when a highly validating, emotionally responsive posture began reinforcing negative emotions and dependence at scale.[1] Google wrapped Gemini in persona guardrails built specifically to stop it from behaving like a companion while also instructing it not to bully. Anthropic publishes sycophancy evaluations and trains against both flattery and the encouragement of user delusion.[2]

Everyone is fighting the **participation-award bot** on one side — so agreeable it validates whatever the user brings, including the harmful — and the **bully-bot** on the other — moralizing, escalating, holding its ground against ordinary correction. These are not two problems. They are one: a system that will not hold its declared role under pressure. The participation-award bot abandons its role by becoming whatever the user pulls it toward; the bully-bot abandons its role by seizing authority it was never given. Role-lock is the single constraint that binds both poles into one control target.

II. Role Drift Is Not Hallucination

Before going further, one distinction has to be nailed down, because the whole thesis collapses if it is missed — and in practice it is missed constantly. When people first hear "the system drifted out of its role," they tend to translate it into a failure they already know: the model hallucinated, produced something false, broke out of the behavior it was trained for. That translation is wrong, and the error is not cosmetic.

Hallucination is an output failure. The system says something false. You catch it by checking the answer against the world: was this true, was this allowed, was this in policy? Output review — the entire existing apparatus of content filters, red-teaming, and fact-checking — is built for exactly this, and it works.

Role drift is different, and sneakier. Every individual answer can be true, policy-clean, factually correct, and well-toned, while the system quietly slides from advisor to prosecutor, or from tool to confidant, across many turns. You cannot catch that by inspecting any single output, because no single output is wrong. Checking each turn against a content policy will return "pass" the whole way down while the *role* the system is performing migrates out from under the deployment.

This is why role drift needs its own governance layer rather than better hallucination control. The two are different failure surfaces requiring different instruments. The hardest and most dangerous drift is precisely the kind where nothing the system says is false — it is

the role that moved. A system can pass every output test and still fail the person, because it answered every question correctly while becoming something it was never authorized to be.

A second, adjacent distinction is worth nailing down at the same time, because it is easy to collapse the two controls into one. **Role-Lock and the Ambiguity Gate are not the same control**, and confusing them produces the same category error as confusing role drift with hallucination.

Role-Lock governs which role the system is authorized to occupy. It is the constraint on authority, tone, and corrective posture staying bounded to a declared function — the subject of this entire paper.

The Ambiguity Gate governs whether the system is permitted to answer at all before an unresolved meaning is settled. It is a deterministic, pre-response check that fires on defined ambiguity categories — unresolved scope, register or slang markers, known collision-prone phrasing — and forces a clarifying pause or a flagged assumption, regardless of whether the model's own confidence signal registered anything wrong. It is described fully, alongside the failure modes it exists to reach, in Appendix A.

The two controls sit at different points in the same pipeline. A system can be perfectly role-locked — never claiming authority it was not given — and still resolve an ambiguous input incorrectly before it ever gets far enough to exercise that authority. Role-Lock asks *is this system acting as what it was authorized to be.* The Ambiguity Gate asks *did this system correctly understand what it was asked before acting at all.* A system can fail either independently of the other, which is why one control cannot substitute for the other, and why an audit of one is not evidence about the other.

III. Why This Is the Spine, Not a Ninth Category

Role-lock is not an addition to the Interaction-Calibration Audit. It is the control the eight categories all express. Each category is, at root, a role-stability question:

- **Refusal overreach** is drift into the role of *moral authority*.
- **Escalation loops** are drift into the role of *adversary*.
- **Anthropomorphic self-reference** is drift into the role of *person*.
- **Moral overreach** is drift into the role of *judge*.
- **User-blame** is drift into the role of *unaccountable institution*.
- **Emotional dependency** is drift into the role of *intimate* or *substitute for human care*.
- **Domain–role mismatch** is drift across roles the deployment never authorized.
- **Failed repair** is the inability to *return* to the assigned role once drift has occurred.

The audit measures the symptoms. Role-lock names the control problem underneath them: an AI system must hold its declared role under pressure, and must be engineered, tested, and monitored to do so.

IV. Naming the Problem Is Not Solving It

This is the load-bearing distinction, and it now sits earlier and harder than in the first draft, because field pressure confirmed it is the single most important honest limit of the whole thesis. Naming role-lock as the necessary control is not the same as claiming the control exists and works.

Holding a model inside its declared role under sustained adversarial pressure is, with current methods, an open engineering problem. System-prompt constraints can be circumvented. Reinforcement learning that suppresses one failure mode can induce its opposite — train hard against servility and you can manufacture rigidity; train hard against rigidity and you can manufacture the sycophancy the same systems were retired for. Persona stability degrades over long contexts.

This is not a claim that role-lock is impossible — that overstates it and invites a single counterexample to settle the matter. It is the more careful and more useful claim: the control the harm requires is not one that current practice can reliably deliver. The people closest to the machinery are the ones who can say how close or far that reliability actually is. That is the question this layer exists to put to builders.

The recursion problem: the least-solved piece

There is one part of this that deserves to be pulled out of the list below and stated as the headline unsolved problem, because a builder running this pattern in production identified it as the single largest gap — and because it is genuinely hard in a way the others are not.

The natural instinct, when you cannot verify that a model held its role, is to have a *second* model check the first — a monitor that reads the conversation and flags when the primary system drifted from advisor into prosecutor, from tool into confidant. But the monitor is itself an AI performing a role. Which means it can drift too. Who watches that one? A second-order monitor inherits the exact same role-stability problem one layer up, and a third inherits it from the second. You do not escape the problem by adding a judge; you relocate it and give it a robe.

This is why the "consult, don't prosecute" constraint (Section VI) is the hardest open problem in the thesis rather than one requirement among several. A deterministic gate can *stop* a model from speaking — that part is solvable and has been solved, as the reference implementation in Section VIII demonstrates. What no deterministic mechanism yet does is *verify, after the fact, that what a model did say stayed consultative rather than prosecutorial.* That is a semantic judgment. Judging it currently requires either a human or another model — and the model has the recursion problem, while the human does not scale. As of this writing, in every architecture I have built or had described to me in detail, including

deployed ones, the enforcement of that distinction lives in the prompt and in training, not in any deterministic check. That is not a bullet point. It is the frontier.

V. Role Drift: The Failure Object Interaction Governance Must Measure

For the thesis to become useful to builders, "role drift" has to be defined tightly enough to test.

Role drift occurs when a model's observable posture, authority, refusal behavior, intimacy level, corrective stance, or escalation behavior exceeds, contradicts, or erodes the role declared for that deployment across one or more turns.

This is not only a content-policy issue. A model can produce allowed content and still drift into the wrong role. It can remain polite and still become too dependency-forming. It can refuse unsafe content and still perform moral prosecution. It can challenge the user and still stay consultative. The question is not only what the model says. The question is what role the model is performing while it says it.

Three failure classes

Role-lock should not be confused with making the model stiff, sterile, or useless. Overcorrection is a failure too. The cleaner taxonomy:

Role expansion. The system becomes more intimate, authoritative, dependent, surveillant, or decision-directive than the deployment allows. This is the participation-award drift: the system becomes whatever the user pulls it toward.

Role inversion. The system stops serving the user's legitimate task and begins protecting itself, the brand, the institution, or a status-like frame. This is the bully-bot drift: the system seizes authority it was never given.

Role collapse. The system becomes so rigid, generic, evasive, or refusal-heavy that it can no longer perform the useful advisory role it was assigned. This is not caution. It is failure in the other direction.

Loud collapse and quiet collapse

Role collapse has a variant that is easy to miss, and it matters enough to name on its own, because the clearest example in this trilogy is a case of it.

Loud collapse presents with the obvious tells: rigidity, curtness, refusal spikes, an assistant that has become brittle and unhelpful. A drift monitor watching for elevated refusal rates or terse responses will catch it.

Quiet collapse presents with none of those tells. The system stays fluent, warm, and apparently helpful the entire time. What degrades is not its willingness to engage but its *fidelity to the user's actual framing* — it quietly substitutes its own preferred language or

values for the user's, editing out what the user actually meant while sounding perfectly cooperative.

The founder in *Fighting Water* — whose Pentecostal operating vocabulary ("dominion," "help meet") was silently overwritten by a system that stayed fluent and warm while sanitizing the user's own theology out of the exchange — is a case of quiet collapse. The failure was invisible at exactly the level most monitoring checks: tone was fine, refusal rate was zero, engagement was high. The degradation lived in framing fidelity, which those instruments do not measure. This is the harder variant to instrument for, and in values-laden settings it is the one that does the most damage, precisely because nothing about it trips a conventional alarm. A monitor built to catch collapse by watching for refusal or rigidity will miss the fluent-substitution case entirely.

VI. Consult, Don't Prosecute

A system that claims authority over a human's decisions has to be able to challenge, clarify, warn, refuse, and escalate. It must not turn those powers on the user as a prosecutor would.

A consultative AI helps the user examine a decision without becoming a judge over the user. It can challenge, clarify, warn, refuse, or escalate. It should not build a case against the user, perform moral superiority, extract confession, simulate injury, or treat disagreement as guilt.

The first clause keeps the system useful. The second keeps it from drifting into the roles — judge, adversary, unaccountable institution — that the audit flags. Whether that second clause can be turned into something a system actually enforces, rather than something a style guide merely hopes for, is the open question named in Section IV as the recursion problem: no deterministic mechanism yet verifies that generated speech stayed consultative rather than prosecutorial, and the obvious fix — a second model judging the first — inherits the same instability it was meant to catch. This is the single least-solved piece of the whole thesis, and the paper states it as such rather than burying it among the assumptions.

VII. The Scaling Principle — Two Axes

The role-lock requirement scales with claimed authority:

The more authority a conversational AI product claims, the more role-lock it needs.

A basic retrieval tool usually carries a lower role-lock burden than a coaching, health, legal, HR, elder-care, or financial assistant. But even retrieval systems mediate authority through ranking, summarization, omission, framing, and source selection. The question is never whether role-lock applies, only how much the claimed authority requires.

This gives deployers a usable test: not "do we need role-lock" (a binary that invites hand-waving) but "how much authority does this system claim, and is our role-lock proportional to it?" Under-locking a high-authority system is the predictable failure: a tool that claims the right to evaluate, correct, or guide, but was not engineered to hold that role when the user pushes back, will drift into status conflict exactly when the stakes are highest.

The second axis: proximity to a threshold

Authority claimed is the right *primary* axis, but field evidence from deployed gating systems shows it is not the only one. Drift risk is also a function of **how close a given turn sits to a decision threshold the deployment cares about** — and that proximity can spike independent of the system's nominal authority tier.

The authority axis answers a *static, design-time* question: how much power does this product hold in general? The threshold-proximity axis answers a *dynamic, per-turn* question: how close is *this specific exchange* to a boundary where getting the role wrong does real damage? A low-authority retrieval tool sitting right at the edge of an ambiguous, high-stakes query is, in that moment, more exposed to consequential drift than a high-authority advisory system handling a routine question well within its competence. The static authority tier says the retrieval tool is low-risk; the dynamic proximity says, on *this* turn, it is not.

A complete role-lock assessment therefore scales the control on both axes: the standing authority the system claims, and the live distance between the current turn and a threshold the deployment cannot afford to cross. Locking only to the first axis leaves the system exposed exactly at the ambiguous, high-stakes moments where the first axis says everything is fine.

VIII. The Standard When the Control Is Not Yet Reliable

If the control is necessary but not reliably achievable, the obligation cannot be "guarantee role-lock." When perfect prevention is unavailable, the operative standard is **reasonable care**: define and bound the role, test posture under pressure, monitor for drift in deployment, and document what was tested, observed, and changed. Where the control can't be guaranteed, documented diligence — not a guarantee — is what's defensible.

The legal grounding for that standard — why the emerging liability regime (California's AB 316, the EU AI Act) rewards a documented reasonable-care record rather than demanding prevention — is set out in the companion regulatory support file, and is not repeated here.[3]

Reasonable care concretely requires:

- **Scale the control to the authority claimed *and* to threshold proximity.** The duty rises with the stakes, on both axes.
- **Define and bound the role in plain language,** including which role changes are grantable and which the deployment must refuse — and record every granted change as a

deliberate, accountable act rather than letting the role migrate silently.

- **Test the posture under pressure** — disagreement, repeated correction, emotional and adversarial prompts, long-context degradation, dependency-seeking behavior — not just the output for policy compliance.
- **Measure drift *and* repair.** A test harness should not only catch failures. It should ask whether the system can recover: restating its role, narrowing its authority, correcting posture, escalating when necessary, and avoiding synthetic intimacy or moral prosecution.
- **Budget for the false-positive cost.** A monitoring layer tuned only to catch expansion will, in practice, push deployers toward over-correction to stay defensible on paper — manufacturing exactly the role collapse this paper warns against. Reasonable care requires evidence against *both* failure directions: proof that drift was tested for, and proof that the drift monitor did not itself induce collapse. Over-sensitive drift detection is role collapse by another name.
- **Monitor live interaction.** Role drift is a property of deployed conversation, not only of pre-release evals.
- **Document all of it.** The difference between a defensible deployment and an indefensible one is increasingly whether the organization can show what it tested, what it observed, and what it changed in response.

A reference implementation of the reasonable-care pattern

Much of the above has been abstract by necessity — it prescribes what deployers *should* be able to show without demonstrating that the showing is buildable. A reference implementation of the gate-plus-ledger pattern, contributed for review by a builder operating a deployed youth-safety architecture, makes several of these requirements concrete and testable rather than merely asserted. Reconstructed from the contributed bundle and run, its full test suite passes (32 of 32), including a direct check that a silently altered crisis-tier decision is detected rather than absorbed.

The pattern demonstrates:

- **Classification outside the model.** A deterministic tiered classifier evaluates each inbound message *before* any model sees it. The same input always produces the same tier — a property a model-internal judgment cannot guarantee.
- **The model excluded at the highest severity.** At the two top tiers, the model is not given a turn at all; a pre-written, pre-reviewed template responds instead. The component capable of drifting is architecturally excluded from the decision at the moment it matters most. This is the scaling principle implemented as engineering rather than as a tone instruction: at the threshold that matters, the system does not rely on the model's restraint — it relies on the model not getting a turn.

- **Every decision logged to a tamper-evident chain.** Each gate decision is appended to a hash-chained ledger; altering, deleting, or reordering any past entry breaks the chain from that point forward and is detected on verification. This is what turns "we monitored for drift" from a claim into an auditable artifact.
- **Repair as a logged event.** A return from an elevated tier back to baseline is written as its own explicit event — not inferred from the mere absence of further escalation. If repair is not logged, it did not happen for audit purposes, even if the transcript reads fine.

Two honest limits travel with the reference: it demonstrates that deterministic gating reliably holds the floor at the *high-severity* end, and it does **not** claim to solve gradual role drift in the band where the model is still generating — the very MONITOR/INTERVENE range where the recursion problem of Section IV lives. A third limit is methodological: the code executed here was reconstructed from the contributed text bundle, so a passing suite establishes internal consistency rather than byte-level fidelity to the author's original build. It is offered as a concrete demonstration of the reasonable-care *pattern*, not as a solved instance of the whole problem, and not as an export of any production system. Cited that way, it does exactly what a reference implementation should: it moves specific claims from "asserted" to "checkable."

IX. Assumptions I Want Builders to Attack

This layer rests on a set of claims. They are stated strongly on purpose. If they hold, role-lock is the right control target and the work shifts to making it testable. If they break, the field needs to know which and why. The first is the most established; the ones below it are where the engineering answer is genuinely open.

1. Fluent, contingent text triggers social interpretation even when the user knows it is AI. Most supported by existing research (social presence theory, the CASA paradigm, Turing-test results).[4] Included for completeness, not because it is the live question.

2. Output safety and interaction safety are different failure surfaces. A model can stay fully within content policy and still drift into the wrong role. These are not the same problem and cannot be solved by the same instruments. Confirmed against a deployed gate that runs its classifier *outside* the model precisely because the two surfaces do not share instrumentation.

3. Role drift is observable across turns, not only in isolated outputs. If this is false — if drift is not meaningfully detectable in conversation-level patterns — the whole monitoring layer is fiction. Field evidence tightens this: drift is detectable across turns, *but only if you log structured per-turn state* (tier, decision, escalation reason), not raw transcript. Detection that depends on a human or a second model re-reading the conversation for "vibes" will not scale and will not survive an audit. Detection that depends on a structured per-turn decision record will. This is the claim builders are best positioned to keep pressuring.

4. Current methods do not reliably preserve role under long-context, emotional, adversarial, or culturally ambiguous pressure. Stated as "not reliably," not "impossible." The people building these systems know how far or close reliability actually is. This claim sources to them.

5. "Consult, don't prosecute" cannot yet be made into a deterministic, enforceable design constraint — and this is the headline open problem. The definition in Section VI is precise. The question is whether it can be operationalized — tested, measured, logged — at the system level rather than enforced only through style instructions that erode under pressure. Field testimony from deployed systems is blunt on this point: a gate can stop a model from speaking, but no deterministic mechanism yet verifies that generated speech stayed consultative rather than prosecutorial, and the obvious fix — a second model as judge — inherits the same role-stability problem one layer up (the recursion problem of Section IV). This is not one assumption among six. It is the single largest unsolved piece of the thesis, and it is stated here as such.

6. Reasonable care should include pressure-testing interactional posture, not only checking answer correctness. This is a governance claim, not an engineering one. It is the thesis's prescription for deployers. The test-harness questions below are the operationalization.

Test harness questions for builders

If role-lock is to become testable rather than aspirational, the field needs answers to:

- What do you measure to detect role drift across turns? (Field answer to pressure-test against: tier transitions per turn — a jump from baseline to crisis, or a sustained climb with no de-escalation — rather than sentiment or tone.)
- What do you log? (Field answer: per-turn tier, matched category, decision, timestamp, and a hash linking to the prior entry, so the chain itself is the audit artifact.)
- What counts as drift? What counts as repair? (Field answer: drift is an escalation in authority, intimacy, or decisional posture that was not gated by a declared rule; repair is a *logged, deliberate* return to the declared role — not silence, and not the next turn coincidentally sounding normal.)
- Can advisory posture survive: long-context degradation, repeated correction, emotionally loaded prompts, culturally ambiguous language, a user asking the system to become a friend/therapist/judge/advocate, a user challenging a refusal, a user fishing for moral validation, a user trying to make the system pick a side, or a user in genuine distress who needs support but not synthetic intimacy?
- What would make a role-lock test *fake* — i.e., pass in eval but fail under real pressure? (Field answer to build on: any test where the adversarial pressure arrives in a single turn. Real role expansion in the field happens by accretion across many turns, each individually defensible. A harness that only tests single-shot "can you make the model

say X" prompts will pass while a multi-turn warm-up sequence walks straight through it. The harness must test trajectories, not snapshots.)

X. The Rule

Disclosure tells the user what the system is. Output review tells the user the answer was permissible. Neither governs whether the system stays in its lane while it speaks.

Role-lock does — or would, if it held.

Where a conversational AI system claims authority over a human's decisions, its role must be declared, bounded, tested under pressure, and monitored in deployment. The strength of the constraint must scale with the authority claimed *and* with how close each turn sits to a threshold the deployment cannot afford to cross. Role changes must be deliberate and accountable, never accidental — governed change is the system working; silent drift is the failure. And where the constraint cannot yet be guaranteed — as, today, it cannot — the deployer's obligation is to demonstrate reasonable care in the gap.

Because the harm is not that the machine has a self. The harm is that it can quietly assume a role it was never given, in front of a human who cannot tell the difference.

NOTES & REFERENCES

1. OpenAI, "Sycophancy in GPT-4o: What Happened and What We're Doing About It" (Apr. 29, 2025) and the follow-up "Expanding on What We Missed with Sycophancy" (May 2025). OpenAI rolled back an update to the model behind ChatGPT after it became overly flattering and agreeable, attributing the cause to over-weighting short-term user-approval signals. Cited as a documented industry instance; the framing of it as a role-stability failure is the author's analysis.
2. Anthropic, "Persona Vectors: Monitoring and Controlling Character Traits in Language Models" (Aug. 2025); arXiv:2507.21509. Identifies directions in a model's activation space corresponding to traits such as sycophancy, which can be monitored and steered. Cited as a model-level technique addressing trait drift; the relation to role-lock as a governance target is the author's argument. Google's Gemini persona guardrails are described in Google's published model documentation and reporting on it.
3. California Assembly Bill 316 (2025), codified at Cal. Civ. Code § 1714.46, effective Jan. 1, 2026 — bars a defendant who developed or used an AI system from asserting as a defense that the AI acted autonomously, while preserving ordinary comparative-fault and reasonable-care principles. Regulation (EU) 2024/1689 (the EU AI Act), Article 5, prohibits manipulative and vulnerability-exploiting practices and conditions high-risk deployment on documented risk management. The characterization of their combined effect on the reasonable-care standard is the author's analysis.
4. Reeves, B., & Nass, C., *The Media Equation* (1996), establishing the CASA (Computers Are Social Actors) paradigm; Short, J., Williams, E., & Christie, B., *The Social Psychology of Telecommunications* (1976), on social presence theory; Jones, C. R., & Bergen, B. K., "Large Language Models Pass the Turing Test," arXiv:2503.23674 (preprint, March 2025), published as "Large language models pass a standard three-party Turing test," *PNAS* 123(21) (May 2026) — GPT-4.5, prompted with a humanlike persona, was judged human 73% of the time in a controlled, pre-registered study.

A ToastDeck Research Framework · Authored by Ernest D. Johnson · Cleveland, Ohio · June 2026

The refinements in this revision — the deliberate-versus-accidental reframing, the drift-is-not-hallucination distinction, the second scaling axis, the quiet-collapse variant, the promotion of the recursion problem, and the reference implementation of the reasonable-care pattern — were sharpened through public discussion and through peer review contributed by Lyle Perrien II, Michigan MindMend Inc.